Indonesian Journal of Electrical Engineering and Computer Science
Vol. 40, No. 2, November 2025, pp. 801∼813
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v40.i2.pp801-813

Monocular vision-based visual control for SCARA-type robotic arms: a depth mapping approach

Diego Chambi[1], Bryan Challco[1], Jonathan Catari[1], Walker Aguilar[1], Lizardo Pari[1]
Electronic Engineering Professional School, Faculty of Production and Services,
Universidad Nacional de San Agustín de Arequipa, Arequipa, Perú

Article Info

Article history:

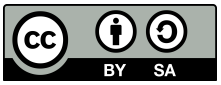
Received Jan 22, 2025
Revised Jul 14, 2025
Accepted Oct 14, 2025

Keywords:

Computer vision
Robotic arm
SCARA-type robotic arm
Vision transformers
Visual servoing

ABSTRACT

The accelerated growth of an increasingly automated industry requires the use of autonomous robotic systems. However, the use of these systems commonly requires an enormous amount of sensors. In this paper we evaluate the performance of a new system for visual control of a selective compliance assembly robot arm (SCARA) robotic arm using a monocular depth map that only requires one monocular camera. This system aims to be an efficient alternative to reduce the number of sensors in the robotic arm area while maintaining the effectiveness of traditional vision algorithms that use stereoscopic architectures of cameras. For this purpose, this system is compared with representative state-of-the-art vision algorithms focused on the control of robotic arms. The results are statistically analyzed, indicating that the algorithm proposed in this research has competitive performance compared to state-of-the-art robotic arm visual control algorithms only using a single monocular camera.



Corresponding Author:

Diego Chambi
Electronic Engineering Professional School, Faculty of Production and Services
Universidad Nacional de San Agustin de Arequipa
04001 Arequipa, Perú
Email: dchambitu@unsa.edu.pe

1. INTRODUCTION

Automation has been employed in every industry in recent years. From precision industrial robots to home automation (domotics), automation has taken on an essential role in performing repetitive and dangerous tasks, allowing humans to focus on activities of greater relevance [1]. Among the many systems used in automation, robotic systems are the most widely adopted and offer the broadest range of applications. These systems were first introduced in factories in the 1960s and, by the 1980s, were being used globally—particularly in the automotive sector. Today, robotic systems are found in a wide variety of settings, including small businesses, educational institutions, and agricultural fields [2]-[4]. Robotic arms, in particular, are composed of multiple links and actuators, enabling them to be used in tasks such as painting, pharmaceutical production, and welding in assembly lines [5]-[7]. Each robotic arm is designed and implemented according to the specific requirements of the task it is intended to perform. To achieve this level of adaptability and precision, robotic arms often require a large number of sensors [8]. Consequently, many industries that could benefit from automation hesitate to adopt robotic systems due to the high cost of these sensing components.

Cameras have been widely used in research on the control of robotic arms; achieving this requires adequate processing of the video captured by the camera. In Intisar *et al.* [9], the video obtained by a camera is processed to classify by color using a transformation to hue, saturation, and value (HSV) in different objects.

Then, a robotic arm performs a pick-and-place task on the selected object. Its interface allows the user to select an object and have it automatically manipulated by the robotic arm without the need for extensive knowledge of the system's inner workings. However, this system directly depends on objects placed at the same level, limiting its functionality. In Kumar *et al.* [10], a stereo camera system generates disparity maps to estimate object location and distance, allowing a robotic arm of three degrees of freedom for pick-and-place tasks. However, it requires precise calibration, synchronization, and computationally intensive algorithms, with added challenges from robotic arm movements in handheld setups. According to Liyanage and Krouglicof [11], visual control for a selective compliance assembly robot arm (SCARA) robot incorporates a high-speed camera with an infrared marker placed at the end effector.

Kim *et al.* [12] highlights a wheelchair-mounted robotic arm that employs stereoscopic cameras along with a coarse-to-fine motion control strategy. As noted in [13], the ARMAR-III robot applies stereo vision combined with stored object orientation data to calculate the full 6D pose of objects relative to their 3D models in real-time, supporting advanced scene analysis. A rose pruning robot, described in [14], integrates stereoscopic cameras positioned near the end-effector to minimize interference. Meanwhile, Ranftl *et al.* [15] discusses a dual robotic arm system that autonomously adjusts the camera's viewpoint to maintain an occlusion-free visual field. Additionally, Urrea and Pascal [16] and Fioravanti *et al.* [17] describe dual-arm systems using stereoscopic cameras for calibration-free control and accurate distance estimation, respectively. Despite these developments, the computational load, sensitivity to environmental changes, and complexity of calibration make stereo vision-based systems impractical for embedded or low-cost applications. Monocular vision algorithms have become a viable substitute in this regard. For instance, Li *et al.* [18] introduces a hybrid visual servo system for agricultural harvesting that uses a single camera, and Nicolis *et al.* [19] investigates the application of Vision Transformers for improved depth prediction in monocular settings. Although these techniques simplify hardware and allow for more flexible deployment, there is still limited integration of these techniques into robotic control systems, especially for pick-and-place and absolute distance estimation tasks.

To fill these gaps, our study suggests a visual control system that integrates a SCARA-style robotic arm with monocular depth estimation based on the MiDaS algorithm [20]. Our suggested method achieves comparable accuracy (RMSE of 0.46 cm) with a single camera, obviating the need for stereo matching and calibration, whereas earlier works like [10], [12], and [13] achieve high precision using stereo vision (e.g., RMSE of 0.49 cm at 15 cm). By making vision-based robotic manipulation more feasible and affordable for embedded systems', where stereo vision systems have traditionally been too costly and computationally demanding, this method tackles important issues. We use a regression-based metric conversion, motivated by [21], to translate the relative depth given by MiDaS into absolute coordinates for robotic control. Inverse kinematics and real-time 3D localization are made possible by this transformation. This system achieves high accuracy in robotic tasks while lowering hardware costs and setup complexity by doing away with the need for stereo cameras. The main contributions of this work are:

- Computational efficiency, the monocular system avoids stereo matching and synchronization overhead [10], [12], enabling its use in low-cost, embedded platforms.
- Precision, RMSE of 0.46 cm at 15 cm, competitive with traditional stereo vision systems (Table 3), providing a high-precision, affordable solution.
- Robustness, stable performance under varying lighting, surpassing baseline systems like [12], making the system more adaptable to real-world conditions.

To the best of our knowledge, this is the first implementation combining i) monocular depth estimation optimized for embedded platforms [20], ii) real-time absolute metric conversion [21], and iii) a low-cost SCARA robotic manipulator manufactured via additive technologies, offering a breakthrough for cost-effective automation in robotics.

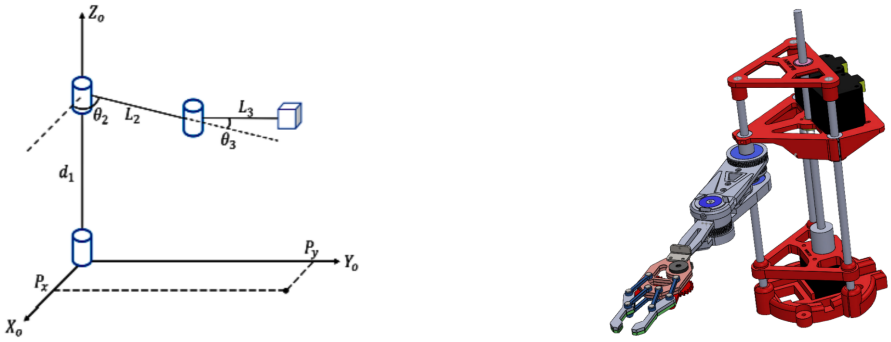
The research is organized as follows: section 2 presents a brief review of the algorithm used for visual control, as well as the materials and methods used to validate the proposed algorithm. Section 3 details the results obtained in the distance estimation and approach tests of the robotic gripper to the target. Section 4 discusses the results highlighting the most relevant observations. Finally, section 5 presents the conclusions and possible lines of future work.

2. METHOD

2.1. Hardware

A 3-DOF SCARA robotic arm was designed and built using additive manufacturing and aluminum rods to validate the proposed vision-based pick-and-place system, given its industrial versatility and ease of control [22], [23]. Previous studies such as [24] have also demonstrated the feasibility of SCARA robots in precision tasks like peg-in-hole assembly, highlighting their suitability for applications requiring accuracy and compliance. To illustrate the structural and analytical basis of the proposed robotic system, Figure 1 shows the proposed SCARA robotic arm's kinematic model and physical structure. Figure 1(a) presents the kinematic configuration, highlighting the three degrees of freedom (d_1, θ_2, and θ_3) and their associated links (L_2, L_3) within a Cartesian reference frame. This model is fundamental for deriving both forward and inverse kinematics. Figure 1(b) shows the CAD rendering of the physical robotic arm, developed through additive manufacturing techniques. This design was optimized in low-cost robotic applications.

The mechanical structure of the SCARA arm was fabricated using PLA for 3D-printed components and aluminum rods for vertical support. The system is actuated by three NEMA17 stepper motors for planar movements and an MG92R servo motor for the gripper. GT2 pulleys and belts are used for motion transmission, while linear bearings ensure smooth movement. The robot is controlled by a GT2560 board programmed using Arduino IDE. The kinematic model of the robotic arm is based on Denavit-Hartenberg (D-H) parameters, which define the spatial relationships between consecutive links. The parameters for each joint are summarized in Table 1. The arm consists of three joints: one prismatic (d_1) and two revolute (θ_2, θ_3). The corresponding link lengths are L_2 and L_3, and all joint offsets are set to zero twist ($\alpha = 0$).



(a) (b)

Figure 1. Proposed SCARA robotic arm's kinematic model and physical structure (a) kinematic representation of the robot with articulated parameters in a Cartesian reference system and (b) physical model of the robot showing its structural design under dynamic conditions

Table 1. D-H parameters of the three D.O.F. for the SCARA robotic arm

	θ	d_i	a_i	α
Joint 1	0	d_1	0	0
Joint 2	θ_2	0	L_2	0
Joint 3	θ_3	0	L_3	0

The kinematics of a serial-link mechanism can be determined through homogeneous transformation matrices, combining basic rotations and translations for each joint, as described by Corke in [25]. Using the Denavit-Hartenberg (D-H) parameters from Table 1, the transformation matrices A_1, A_2, and A_3 are computed. The direct kinematics is obtained by multiplying these matrices:

$$T_3 = A_1 \cdot A_2 \cdot A_3 \tag{1}$$

The resulting matrix T_3 gives the position and orientation of the end effector with respect to the base frame. In its expanded form, the position is a function of the joint angles θ_2 and θ_3, and the link lengths L_1,

L_2, and L_3. To calculate the joint angle θ_3 for object manipulation, the inverse kinematics equation is used:

$$\theta_3 = \arccos\left(\frac{P_x^2 + P_y^2 - L_1^2 - L_2^2}{2L_1L_2}\right) \tag{2}$$

For the vision system, Avatec cameras with a 720p resolution, USB interface, and 30 FPS refresh rate were used to track the position of the object, separately or in stereo configuration.

2.2. Software

This subsection details the algorithms necessary to perform the picking task with the SCARA robotic arm. Typically, stereoscopic vision-based systems use tracking algorithms to obtain a disparity between cameras. To represent this type of system, we implement this algorithm using the MIL tracking model, as discussed in [26]. As a second system, the monocular vision depth mapping algorithm is introduced. In this configuration, only one webcam is used along with the MiDaS model, which has been shown to effectively estimate depth from monocular images [27]. The performance of this visual control system is then compared to the conventional stereoscopic camera system. The two systems to be compared are summarized as follows:

- Stereoscopic architecture: an algorithm based on stereoscopic vision using MIL tracking and the disparity algorithm, as outlined in [26].
- Monocular vision: the proposed system uses the MiDaS algorithm based on monocular vision [27].

Once each algorithm detects the position of the object in the three Cartesian coordinates, a third algorithm based on the kinematics of the robotic arm will pick up the indicated object. A user interface allows the user to signal the object to be picked up by the gripper for manipulation by the robotic arm, as described in [28] and [29].

2.2.1. Stereoscopic architecture

In this vision mode, a two-camera array in stereo configuration is used. This algorithm is widely used in visual control systems for robotic arms due to its simple operating principle. Usually, an object tracking algorithm is used so that the operator can select the object to perform the pick-and-place task with the robotic arm through a user interface. We used the MIL algorithm for this specific case, which is considered one of the most robust against disturbances in continuous image capture. We use the $OpenCV$ library and the command: $Python.TrackerMILcreate$. Once the object is tracked, we obtain the center of mass by obtaining the mass moments 0, 0 using the command $cv2.moments$. We then use the disparity algorithm to calculate the distance between the object and the stereo camera array. Figure 2 graphically shows the disparity obtained from a position difference captured by both cameras.

In Figure 2(a), O_c represents the optical centers of the cameras, T is the baseline, and f is the focal length of each lens. The point P is the object in the environment, and Z is the distance we want to calculate. In Figure 2(b), we observe the object seen by both frames of the stereoscopic camera, where X_L and X_R are the distances from the reference frame of each camera to the center of mass of the detected object.

(a) (b)

Figure 2. Disparity obtained from a position difference captured; (a) depth triangulation scheme in stereo vision showing the geometry of the cameras and the observed object and (b) disparity representation in images captured by left and right cameras to estimate the distance

To calculate the distance Z to the object using stereo vision, the following steps are carried out. First, the positions X_L and X_R of the object are extracted from the left and right camera frames, respectively. The disparity is then calculated as the difference between these two positions:

$$\text{disparity} = X_L - X_R$$

If the disparity is zero (i.e., the object is directly aligned between both cameras), it is adjusted to a small value (usually 1) to avoid division by zero. The depth, or distance Z, is then computed using the formula:

$$Z = \frac{f \cdot T}{\text{disparity}}$$

where f is the focal length of the cameras, and T is the baseline (the distance between the two cameras). The result is the estimated distance Z to the object along the Z-axis. The distance Z is always positive, so its absolute value is taken to ensure the result is non-negative.

In this way, the triangulation process determines the 3D coordinates of the object in space by calculating its location on the X and Y axes, along with the approximate distance to the Z axis.

2.2.2. Monocular architecture

For the proposed monocular vision system, we use the MiDaS depth estimation model based on deep learning. Recent work by Smith *et al.* [30] introduced alternative methods for linear depth estimation from uncalibrated monocular images using polarization cues; however, our approach focuses on transformer-based depth prediction for robotic control applications. MiDaS offers three versions with varying computational demands. To reduce the implementation cost of visual control in industrial robotic arms, we selected the Small version due to its low computational requirements, which makes it suitable for low-power processors.

Figure 3 shows the depth map generated by the MiDaS algorithm and the corresponding top view of the test object. In Figure 3(a), the depth map is visualized with colors that indicate the relative distances of the objects. Figure 3(b) presents the same scene converted to grayscale, highlighting the depth variations more clearly for easier processing by the control system.

(a) (b)

Figure 3. Depth map is visualized with colors that indicate the relative distances of the objects (a) MiDaS small algorithm example and (b) proposed image processing for monocular architecture.

The monocular vision system uses a neural network based on backbones for distance estimation. However, applying it to industrial robotic tasks requires additional signal processing steps, including perspective transformation, noise filtering, and absolute distance estimation from relative measurements. Figure 4 summarizes these sequential steps, which are detailed below.

First, the image of the webcam is captured; this video, obtained from a single camera, presents a "fisheye" effect that spherically distorts the image. To correct for this distortion, a perspective transformation is performed using the command $cv2.warpPerspective$, which requires selecting four points at the edge of the working area. Once the image has been corrected, the MiDaS depth map algorithm is applied, specifically selecting the Small version. This model is loaded from the $Pytorch$ library using the command $midas = torch.hub.load('intel-isl/MiDaS', MiDaS_small)$. At this stage, a depth map version of the input image is obtained. Subsequently, the depth map is normalized using the $cv2.normalize()$ command; for this research, normalization was applied to a range between 1 and 10 to facilitate further data processing. Figure 3(b) shows an example of this normalized depth map in the robotic arm's workspace.

Figure 4. Sequential steps for absolute distance estimation

After normalization, spline interpolation is performed to smooth transitions between pixel values. From the interpolated data, the distance from the center of the tracked object to the camera is calculated. A moving average filter is then applied to stabilize the obtained values over time. Subsequently, the relative distance between the background and the object is determined based on the generated depth map. In Masoumian *et al.* [24], a similar problem is addressed by approximating the absolute distance from the relative measurement using a quadratic function given by:

$$Y = (c_0 + c_1 X + c_2 X^2)h \tag{3}$$

where the coefficients C_0 , C_1 and C_2 are obtained using least squares equations, h is the height at which the chamber is located, and X is the relative distance. This same problem is presented in [25] and is solved by finding the optimal curve through least squares equations. For this, a total of six images at different distances from the camera were used to calibrate the model. Finally, the estimated absolute distance is subtracted from the 35 cm height at which the camera is located to determine the object's height.

The core steps of the monocular vision and distance estimation process are summarized in Algorithm 1. The algorithm follows these steps: First, the image is captured and the perspective distortion is corrected. Then, the depth map is generated and normalized, followed by distance calculation. Finally, the absolute distance is estimated using a quadratic fitting model.

Algorithm 1 Proposed algorithm

```
procedure PERSPECTIVE_CORRECTION(frame)
    P1, P2, P3, P4 ← select four points
    Points ← [P1, P2, P3, P4]
    if length(Points) = 4 then
        new_frame ← cv2.WarpPerspective(frame, Points)
    end if
    frame ← new_frame
end procedure

procedure DEPTH_MAP(frame, img_batch)
    Midas ← model_type.MiDaS_small
    depth_map ← Midas(img_batch, frame)
    depth_map ← depth_map.interpolate(frame)
    depth_map ← cv2.normalize(depth_map)
end procedure

procedure DISTANCE_TO_CAMERA(frame, depth_map)
    Tracking_algorithm ← MIL
    Object ← select.Object
    [XC, YC] ← Tracking_algorithm(Object)
    Bounding_box ← Tracking_algorithm(Object, frame)
end procedure

procedure ABSOLUTE_DISTANCE_ESTIMATION(frame, Relative_distance)
    x ← [11.8, 10.843, 10.411, 10.2]
    y ← [21, 23, 26, 31]
    degree ← 2
    Quadratic_function ← np.polyfit(x, y, degree)
    Distance ← Quadratic_function(Filtered)
end procedure
```

To provide a practical demonstration of the entire process, a video has been included that shows the monocular vision system in action with the SCARA robotic arm. The video illustrates how the steps outlined

in Algorithm 2 are executed, from image capture and perspective correction to depth map generation and object tracking. This visual example helps to clarify the methodology and highlight the system's functionality. The video can be viewed at the [31].

Algorithm 2 Robotic arm control

procedure MICROCONTROLLER(SerialCommunication)
  $Motor1Step, Motor1Dir \leftarrow 25, 23$
  $Motor1Angle\_gets(200/360) \leftarrow (62/20)$
  $Motor2Step, Motor2Dir \leftarrow 31, 33$
  $Motor2Angle\_gets(200/360) \leftarrow (89/20)$
  $MotorZStep, MotorZDir \leftarrow 37, 39$
  $Motor1Distance \leftarrow 200/1.2$
  $ServoMotorPin \leftarrow 11$
  $[M1, M2, Mz, Servo] \leftarrow$ SerialCommunication
  $Motor1Position \leftarrow M1 * Motor1Angle$
  $Motor2Position \leftarrow M2 * Motor2Angle$
  $MotorZPosition \leftarrow MZ * Motor1Angle$
  $ServoMotorPosition \leftarrow Servo$
end procedure
procedure INVERSE KINEMATICS((p_x, p_y, p_z, SpaceButton))
  $d_1 = p_z$
  Gripper $\leftarrow 180$
  $\theta_3 \leftarrow \arccos\left(\frac{p_x^2+p_y^2-l_1^2-l_2^2}{2l_1l_2}\right)$
  $\theta_2 \leftarrow \left(\frac{l_2(p_x*\sin(\theta_3)+p_y*\cos(\theta_3))+p_y l_1}{p_x^2+p_y^2}\right)$
  data $\leftarrow [\theta_2, \theta_3, d_1,$ Gripper]
  if SpaceButton $= 1$ **then**
    SerialCommunication $\leftarrow$ data
  end if
end procedure

2.2.3. Robotic arm control

Once the object is fixed and its exact position has been obtained through the algorithms detailed above, the inverse kinematics of the robotic arm are used so that it reaches the object and picks it up. In algorithm 3, the first procedure corresponds to the algorithm implemented in the microcontroller of the robotic arm, which is in charge of receiving through serial communication the data of the angles that each motor must travel; for this, we must transform the steps that the motor must take to the necessary angle considering the teeth of the motor gear and the pulley of the corresponding link. Within this microcontroller procedure, we also need to name the pins connected to the motors for control obtained from the GT2506 board.

For the case of the motor that raises or lowers the robotic arm in the Z axis, the transformation is reduced as follows:

$$\text{AngleToSteps} = \frac{\text{BeltTeeth}}{\text{GearTeeth}}$$

The second procedure presented in the algorithm represents the inverse kinematics that is executed in the computer that has serial communication with the robot, this calculation is given by the equations calculated in the Hardware subsection. Finally, a conditional waits for the operator's indication by pressing the space key for the degrees to be sent by serial communication to the microcontroller and executed by the robotic arm.

3. RESULTS

3.1. Results on distance estimation

Each system was tested with the SCARA robot, using additively printed objects at different heights and positions within the workspace. The system includes a low-cost SCARA robot, a laptop for processing, test objects, a monocular camera, and a stereoscopic camera array. The complete setup, including all components, can be seen in Figure 5, which shows both the hardware and the arrangement of the sensors and robotic arm in the test environment.

Figure 5. Setup implemented for experimentation

Multiple picking tasks are performed with each algorithm to evaluate both proposed systems. Because we seek to implement a system that correctly identifies the position of the object so that the robotic gripper can pick it up, we do not consider evaluating parameters such as speed, torque, or power consumption of the robotic arm. In addition, a user interface was implemented that allows the user to select the object to be picked up with the robotic arm. Within this interface, the user can see the camera view in real time and select the objects to be picked up by the robotic arm; for this experiment, circular figures were used in both cases to make a fair evaluation.

Table 2 presents the estimated distances and corresponding error values obtained using both the proposed monocular vision system and a traditional stereoscopic system. The table is divided into two main groups: one for real distances of 15 cm and 13 cm (left side) and another for 10 cm and 5 cm (right side). Each group compares the estimated distance with the actual object distance, and the difference is shown as the estimation error. A color-coded heatmap highlights low (green), moderate (yellow), and high (red) errors, facilitating a visual assessment of accuracy. This format allows for a clear comparative analysis between the two systems across multiple trials and distances.

Table 2. Distance estimation at 15 cm - 13 cm and 10 cm and 5 cm

Real Distance (cm)	Monocular vision Proposed		Stereoscopic		Real Distance (cm)	Monocular vision Proposed		Stereoscopic Vision	
	Estimated Distance	Error (cm)	Estimated Distance	Error (cm)		Distance	Error (cm)	Estimated Distance	Error (cm)
15	14.809	0.191	14.268	0.732	10	10.654	0.654	9.842	0.158
15	14.854	0.146	15.573	0.573	10	9.334	0.666	10.369	0.369
15	14.760	0.240	14.675	0.325	10	10.412	0.412	11.019	1.019
15	14.643	0.357	15.294	0.294	10	10.688	0.688	9.738	0.262
15	14.434	0.566	14.921	0.079	10	10.101	0.101	10.124	0.124
15	14.112	0.888	15.407	0.407	10	10.838	0.838	10.965	0.965
15	14.978	0.022	14.351	0.649	10	10.928	0.928	9.173	0.827
15	14.225	0.775	15.831	0.831	10	10.263	0.263	9.512	0.488
15	15.393	0.393	14.733	0.267	10	10.978	0.978	10.876	0.876
15	15.094	0.094	15.122	0.122	10	10.145	0.145	9.321	0.679
15	13.133	0.133	12.367	0.633	5	5.316	0.316	4.825	0.175
13	13.484	0.484	13.721	0.721	5	5.755	0.755	5.692	0.692
13	14.087	1.087	12.946	0.054	5	5.583	0.583	4.213	0.787
13	13.224	0.224	14.012	1.012	5	5.086	0.086	5.336	0.336
13	12.235	0.765	13.532	0.532	5	5.557	0.557	4.181	0.819
13	13.389	0.389	12.689	0.311	5	5.782	0.782	5.812	0.812
13	12.791	0.209	13.248	0.248	5	5.805	0.805	4.567	0.433
13	13.031	0.031	12.574	0.426	5	4.923	0.077	6.109	1.109
13	13.804	0.804	13.896	0.896	5	5.034	0.034	4.896	0.104
13	13.114	0.114	12.315	0.685	5	4.702	0.298	4.429	0.571

When compared with existing stereo vision systems, such as those described in [10] and [13]—where stereo setups with dual high-precision cameras were used, achieving RMSE values around 0.49 cm at 15 cm—our monocular system achieves comparable accuracy (RMSE of 0.46 cm) while requiring only a single

camera. This makes our approach more cost-effective and easier to deploy, particularly in resource-constrained environments.

These results demonstrate that the proposed monocular system can perform at a level of accuracy similar to that of stereo vision systems, but with far fewer hardware requirements. The implication of this is significant for applications in industrial robotics, where minimizing hardware cost and complexity is often crucial. By replacing expensive stereo vision setups with a single camera, we open up the possibility of implementing visual control systems in more cost-effective and embedded robotic platforms.

Table 3 provides a comparative analysis of the monocular vision algorithm (proposed) and the stereoscopic vision algorithm based on minimum error, maximum error, and root mean square error (RMSE) at different real distances. The results show that the monocular vision algorithm achieves lower RMSE at shorter distances while maintaining competitive performance at longer distances, highlighting its robustness and reliability compared to the stereoscopic method.

Table 3. Comparison of monocular vision (proposed) and stereoscopic vision, error and RMSE

Real distance (cm)	Monocular vision (proposed)			Stereoscopic vision		
	Error max (cm)	Error min (cm)	RMSE (cm)	Error max (cm)	Error min (cm)	RMSE (cm)
15	0.888	0.022	0.4600	0.831	0.079	0.4925
13	1.087	0.054	0.5407	1.012	0.054	0.6189
10	0.978	0.124	0.6430	1.019	0.124	0.6607
5	0.805	0.104	0.5179	1.109	0.104	0.6577

For a comparative view of the results, in Figure 6 is represented the results of Table 2 in a box plot; the results are grouped in pairs, each representing the estimation of the monocular vision algorithm and the stereo vision-based algorithm, being four pairs for the proposed distances.

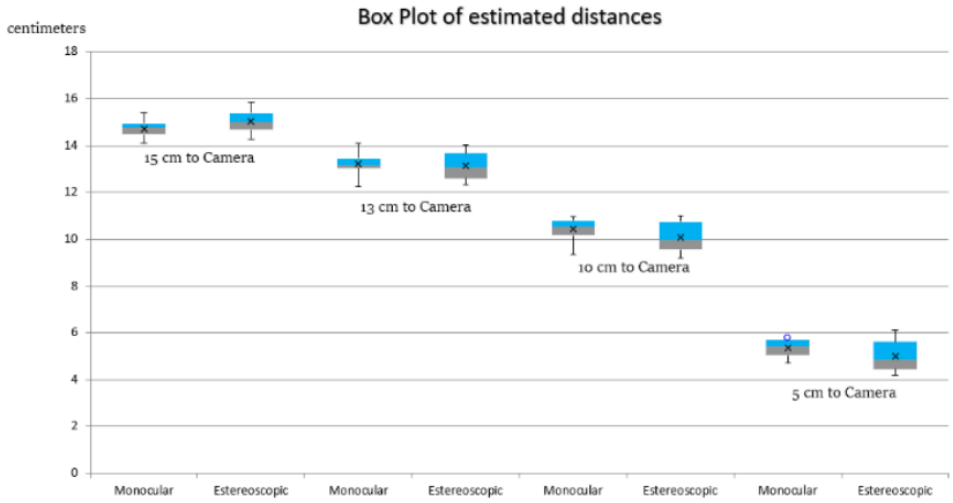


Figure 6. Box plot comparing distance estimation errors

Due that the main focus of the algorithms presented is the determination of the distance of the cameras to the target, a statistical analysis is performed to evaluate the performance of both algorithms in this estimation. From the errors in Table 4, we obtain normal distributions according to the Shapiro-Wilk test. However, there is no homogeneity of variances according to Levene's test; due to this, we use a non-parametric analysis based on the Mann-Whitney U test. The following hypotheses are assumed for this test:

- $\mathbf{H_0}$: There is a significant difference between both groups of data.
- $\mathbf{H_i}$: There are no significant differences between the two data groups.

By assigning a significance value Alpha = 0.05 or 5%, the P values shown in Table 4 are obtained.

Table 4. Hypotheses for each estimation distance.

Distance (cm)	α	¶Value	H_0	H_i
15	0.05	0.09938	**Accepted**	**Rejected**
13	0.05	0.18217	**Accepted**	**Rejected**
10	0.05	0.14495	**Accepted**	**Rejected**
5	0.05	0.11323	**Accepted**	**Rejected**

In summary, the monocular vision algorithm offers considerable benefits in terms of cost and simplicity while exhibiting strong performance with small error margins and reaching accuracy levels that are comparable to stereoscopic systems. These findings suggest that monocular vision can be a very successful substitute for robotic applications, especially in settings where computational efficiency and cost reduction are top priorities. These results validate our initial hypothesis that a monocular vision system can serve as a viable and cost-effective alternative to more complex stereoscopic systems in robotic applications.

3.2. Results on gripper approximation

Once the distances of the objects to the camera are estimated, the results of the approximations of the robotic gripper of the SCARA arm to the position of each object are calculated using the inverse kinematics equations presented in the Hardware section. The calculations performed by Algorithm 2 containing the kinematics equations are shown in Table 5, as well as the errors obtained between the actual position and these calculations. This error is given by the difference between two points in 3 dimensions by the following:

$$\text{Error} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

Table 5. Gripper approximation results with proposed algorithm

Real position			Gripper position			Error
X axis (cm)	Y axis (cm)	Z axis (cm)	X axis (cm)	Y axis (cm)	Z axis (cm)	(cm)
5.00	5.00	15.00	5.013	4.823	14.809	0.2607
5.00	12.50	15.00	4.847	12.422	14.854	0.2254
5.00	20.00	15.00	4.786	20.453	14.760	0.5555
5.00	5.00	15.00	5.074	14.643	14.643	0.3660
12.50	12.50	15.00	12.385	12.493	14.434	0.5776
12.50	20.00	15.00	12.871	19.765	14.112	0.9907
20.00	5.00	15.00	20.122	5.018	14.978	0.1253
20.00	12.50	15.00	19.964	12.405	14.225	0.7816
20.00	20.00	15.00	19.817	19.958	15.393	0.4355
5.00	5.00	10.00	5.056	4.896	10.654	0.6646
5.00	12.50	10.00	5.110	12.506	9.334	0.6750
5.00	20.00	10.00	4.935	19.509	10.412	0.6442
12.50	5.00	10.00	12.578	5.098	10.688	0.6993
12.50	12.50	10.00	12.492	12.381	10.101	0.1563
12.50	20.00	10.00	12.853	19.872	10.838	0.9183
20.00	5.00	10.00	20.262	4.842	10.928	0.9771
20.00	12.50	10.00	20.114	12.485	10.263	0.2870
20.00	20.00	10.00	19.973	20.421	10.978	1.0651
5.00	5.00	5.00	4.823	5.044	5.316	0.3649
5.00	12.50	5.00	5.276	12.519	5.755	0.8041
5.00	20.00	5.00	5.198	20.167	5.583	0.6380
12.50	5.00	5.00	12.622	4.897	5.086	0.1814
12.50	12.50	5.00	12.735	12.365	5.557	0.6194
12.50	20.00	5.00	12.631	20.352	5.782	0.8675
20.00	5.00	5.00	20.255	4.932	5.805	0.8472
20.00	12.50	5.00	20.153	12.460	4.923	0.1759
20.00	20.00	5.00	20.318	20.122	5.034	0.3423

At larger distances (e.g., 15 cm), the gripper's approximation error is relatively small, with a maximum error of 0.2607 cm. However, at shorter distances, such as 5 cm, the error increases to 0.8675 cm, suggesting that the system performs better at longer ranges but needs further optimization for accuracy at close distances. The gripper's approximation errors align with previous studies, which report errors of 0.5 cm to 1 cm for similar robotic systems using inverse kinematics for position estimation at 10 to 15 cm distances [9], [10]. Our system, with maximum errors around 1.0651 cm at 5 cm, shows comparable performance but highlights the potential

for improvement with additional calibration. Finally, Table 6 presents the maximum, minimum, and average errors at distances of 5 cm, 10 cm, and 15 cm.

Table 6. Summary of gripper approximation errors

Z distance (cm)	Min. error (cm)	Max. error (cm)	Average error (cm)
5	0.1759	0.8675	0.5378
10	0.1563	1.0651	0.6782
15	0.1253	0.9907	0.4798

In conclusion, the results show that the monocular vision system achieves high accuracy in robotic control. While the system performs well overall, errors are slightly higher at shorter distances. These results support the use of monocular vision in low-cost robotics, especially for industrial and research applications.

4. DISCUSSION

Based on the distance estimation results, we observe in Table 2 that there is no significant difference between the proposed monocular system and the stereo system in terms of distance estimation. The results of the Mann-Whitney statistical tests confirmed that there are no substantial differences in errors between the two methods. This validates our hypothesis that a monocular system can achieve competitive performance compared to traditional stereo systems, which is significant for robotic applications, as it allows reducing the complexity of the required sensors without sacrificing accuracy. However, the errors in estimating the gripper's position, which reached up to 1 cm, can be due to several factors. These include inaccuracies in the dimensions used in the robotic arm control algorithm, nonlinear behaviors of the system that were not modeled, or inadequate stresses on the prototype components. These errors are consistent with other studies using visual algorithms for robot control, as seen in [10], which also report relatively low errors in position estimation. Despite the slight difference in errors compared to stereo systems, the results of this research highlight that the monocular system can be an effective solution for low-cost robotics applications, significantly reducing the number of sensors required. This opens the door to the implementation of more accessible systems in industrial or research environments where traditional stereo-vision-based systems are expensive and complex to implement.

5. CONCLUSIONS AND FUTURE WORK

The research shows us an innovative approach for robotic arms, specifically in that they can achieve good results in accuracy and calculations using a single lens with the classic stereo vision system, thanks to the tests and statistics where it is concluded that there is no significant difference between the errors obtained with the monocular vision algorithm compared to the stereo vision algorithm, which validates this new system and the entire system is implemented in the SCARA robotic arm where resources are minimized. This research, contributes in the field of robotics and industry, because this vision system reduces costs, simplifies the hardware and seeks to get the most performance using the minimum resources without losing a big difference in quality. However, we also take into account that there are many more opportunities for improvement of optimization such as improvement through image processing and better use of the midas algorithm, not losing the characteristic that are agile processes of low cost, we present an effective solution for robotics and opportunities for improvement in the development of intelligent robots with low resources.

ACKNOWLEDGMENTS

"We would like to sincerely express our gratitude to the National University of San Agustín of Arequipa for their valuable support in providing equipment and access to expert faculty advisors, which has been instrumental in advancing this research."

FUNDING INFORMATION

Authors state no funding involved.

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

Data availability is not applicable to this paper as no new data were created or analyzed in this study.

REFERENCES

[1] M. A. K. Bahrin, M. F. Othman, N. H. N. Azli, and M. F. Talib, "Industry 4.0: A review on industrial automation and robotics," *J. Teknol.*, vol. 78, no. 6-13, Jun. 2016.

[2] J. Seol *et al.*, "Human-centered robotic system for agricultural applications: Design, development, and field evaluation," *Agriculture*, vol. 14, no. 11, p. 1985, Nov. 2024.

[3] L. F. P. Oliveira, A. P. Moreira, and M. F. Silva, "Advances in agriculture robotics: A state-of-the-art review and challenges ahead," *Robotics*, vol. 10, no. 2, p. 52, Mar. 2021.

[4] S. Singh, R. Vaishnav, S. Gautam, and S. Banerjee, "Agricultural robotics: A comprehensive review of applications, challenges and future prospects," in *Proc. 2nd Int. Conf. Artif. Intell. Mach. Learn. Appl. (AIMLA)*, Namakkal, India, 2024, pp. 1–8.

[5] J. M. Gülzow, P. Paetzold, and O. Deussen, "Recent developments regarding painting robots for research in automatic painting, artificial creativity, and machine learning," *Appl. Sci.*, vol. 10, no. 10, p. 3396, May 2020.

[6] K. M. Yusuf and A. Gavit, "Automation and robotics in pharmaceutical industry - Review," *Int. J. Pharm. Res. Appl.*, vol. 9, no. 3, pp. 115–132, May 2024.

[7] Y. Xiong, Y. Ge, L. Grimstad, and P. J. From, "An autonomous strawberry-harvesting robot: Design, development, integration, and field evaluation," *J. Field Robot.*, vol. 37, no. 2, pp. 202–224, Aug. 2019.

[8] Z. Zhao *et al.*, "Toward generalizable robot vision guidance in real-world operational manufacturing factories: A semi-supervised knowledge distillation approach," *Robot. Comput.-Integr. Manuf.*, vol. 86, p. 102639, Apr. 2024.

[9] M. Intisar, M. M. Khan, M. R. Islam, and M. Masud, "Computer vision-based robotic arm controlled using interactive GUI," *Intell. Autom. Soft Comput.*, vol. 27, no. 2, pp. 533–550, 2021.

[10] S. G. R., N. Kumar, H. P. R., and S. S., "Implementation of a stereo vision-based system for visual feedback control of robotic arm for space manipulations," *Procedia Comput. Sci.*, vol. 133, pp. 1066–1073, 2018.

[11] M. H. Liyanage and N. Krouglicof, "An embedded system for a high-speed manipulator with single time scale visual servoing," *J. Dyn. Syst. Meas. Control*, vol. 139, no. 7, 2017.

[12] D.-J. Kim, R. Lovelett, and A. Behal, "Eye-in-hand stereo visual servoing of an assistive robot arm in unstructured environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2009, pp. 2326–2331.

[13] T. Asfour *et al.*, "ARMAR-III: An integrated humanoid platform for sensory-motor control," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, 2006, pp. 169–175.

[14] H. Cuevas-Velasquez *et al.*, "Real-time stereo visual servoing for rose pruning with robotic arm," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Paris, France, 2020, pp. 9284–9290.

[15] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020.

[16] C. Urrea and J. Pascal, "Design, simulation, comparison and evaluation of parameter identification methods for an industrial robot," *Comput. Elect. Eng.*, vol. 67, pp. 791–806, Apr. 2018.

[17] D. Fioravanti, B. Allotta, and A. Rindi, "Image based visual servoing for robot positioning tasks," *Meccanica*, vol. 43, no. 3, pp. 291–305, 2008.

[18] Y.-R. Li, W.-Y. Lian, S.-H. Liu, Z.-H. Huang, and C.-T. Chen, "Application of hybrid visual servo control in agricultural harvesting," in *Proc. Int. Conf. Syst. Sci. Eng. (ICSSE)*,2022, pp. 235–238.

[19] D. Nicolis, M. Palumbo, A. M. Zanchettin, and P. Rocco, "Occlusion-free visual servoing for the shared autonomy teleoperation of dual-arm robots," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 796–803, Apr. 2018.

[20] S. Zuo, Y. Xiao, X. Chang, and X. Wang, "Vision transformers for dense prediction: A survey," *Knowl.-Based Syst.*, vol. 239, p. 109552, 2022.

[21] P. Hynes, G. I. Dodds, and A. J. Wilkinson, "Uncalibrated visual servoing of a dual-arm robot for surgical tasks," in *Proc. IEEE Int. Symp. Comput. Intell. Robot. Autom. (CIRA)*, Espoo, Finland, 2005, pp. 445–450.

[22] A. Masoumian *et al.*, "Absolute distance prediction based on deep learning object detection and monocular depth estimation models," *Appl. Sci.*, vol. 11, no. 19, p. 9334, Sep. 2021.

[23] S. H. Han, W. H. See, J. Lee, M. H. Lee, and H. Hashimoto, "Image-based visual servoing control of a SCARA-type dual-arm robot," in *Proc. IEEE Int. Symp. Ind. Electron. (ISIE)*, 2000, vol. 2, pp. 517–522.

[24] R. K. Jain, S. Majumder, and A. Dutta, "SCARA based peg-in-hole assembly using compliant IPMC micro gripper," *Robot. Auton. Syst.*, vol. 61, no. 3, pp. 297–311, Mar. 2013.

[25] I. Corke, "A simple and systematic approach to assigning Denavit–Hartenberg parameters," *IEEE Trans. Robot.*, vol. 23, no. 3, pp. 590–594, Jun. 2007.

[26] H. Sundaram and G. D. Hager, "Visual tracking with online Multiple Instance Learning," in *2009 IEEE Conference on computer vision and Pattern Recognition*, 2009, doi: 10.1109/CVPR.2009.5206737.

[27] T. Ehret, "Monocular Depth Estimation: a Review of the 2022 State of the Art," *Image Processing On Line*, 2023, doi: 10.5201/ipol.2023.459.

[28] O. D. Faugeras, Q. M. Luong, and T. Papadopoulo, The Geometry of Multiple Images: The Laws That Govern the Formation of Multiple Images of a Scene and Some of Their Applications, *MIT Press*, 2001, doi: 10.7551/mitpress/3259.001.0001.

[29] S. Lahiri, J. Ren, and X. Lin, "Deep Learning-Based Stereopsis and Monocular Depth Estimation Techniques: A Review," *Vehicles*, vol. 6, no. 1, pp. 305-351, 2024, doi: 10.3390/vehicles6010013.

[30] W. A. P. Smith, R. Ramamoorthi, and S. Tozza, "Linear depth estimation from an uncalibrated, monocular polarisation image," in *Computer Vision – ECCV 2016*, Lecture Notes in Computer Science, vol. 9908, pp. 109–125, 2016.
[31] Walker Alexis Aguilar Rodriguez. Monocular Vision-Based Visual Control for SCARA-TypeRobotic Arms: A Depth Mapping Approach. (May 4, 2025). Accessed: Jul. 9, 2025. [Online video]. Available: https://www.youtube.com/watch?v=zwymRfo9bI8.

BIOGRAPHIES OF AUTHORS

Diego Chambi Bachelor's degree in Electronic Engineering from UNSA, with experience in the industrial and mining sectors. He specializes in automation, instrumentation, and process control, with knowledge of safety, occupational health, and environmental management in mining. He possesses solid expertise in electrical rooms, substations, electric motors, generators, industrial instrumentation, structured cabling, working at heights, and electrical panels. As a Bachelor's graduate in Electronic Engineering, he has developed skills to create innovative and effective solutions while working under pressure and in a team. He can be contacted at email: dchambitu@unsa.edu.pe.

Bryan Challco Bachelor's degree in Electronic Engineering seeking professional opportunities in projects and research. He stays updated on the latest technological advancements and aims to work in instrumentation, control, and automation teams to contribute to future generations. He can be contacted at email: bchallco@unsa.edu.pe.

Joanathan Catari holds a Bachelor's degree in Mechatronics Engineering from the Technological University of Peru (UTP), Arequipa, and a Technical degree in Industrial Electronics from the Pedro P. Díaz Higher Institute, Arequipa. A specialist in embedded electronic systems design, IoT, industrial robotics, and automation, he has contributed to scientific instrumentation projects, including bionanosensors and sensor systems for nanosatellites, in collaboration with the National University of Frontera (UNF), Sullana, 3R Aerospace & Nanotechnology, France, and the University of Engineering and Technology (UTEC), Lima. He can be contacted at email:jcatari@unsa.edu.pe.

Walker Aguilar received the B.S. degree in electronic engineering from the National University of San Agustin of Arequipa, Peru, in 2023. His expertise extends as a Hardware Developer for startups in Peru, offering innovative solutions. He is currently a Research Assistant at UNSA in the project "Design and Development of an EOD robot with Multimodal Control Interface"—CONCYTEC. His research interests include robotics and hardware. He can be contacted at email:waguilarr@unsa.edu.pe.

Lizardo Pari received the B.S. and M.S. degrees in electronic engineering from Universidad Nacional de San Agustín de Arequipa, Peru, in 1996 and 2005, respectively, and the Ph.D. degree in automation and robotics from the Polytechnic University of Madrid, Spain, in 2007. He is currently a Principal Professor and the Head of the Vision and Robotics Group, Universidad Nacional de San Agustín de Arequipa. His research interests include computer vision, motion tracking, threedimensional reconstruction, and visual control. He was a member of the IEEE Robotics and Automation Society Peru Section. He can be contacted at email:pparip@unsa.edu.pe.